VERIFICATION OF TRANSLATION

I, Kyozo Omori, translator of 831-9, Ono, Sanda, Hyogo, Japan, hereby declare that I am conversant with the English and Japanese languages and am a competent translator thereof. I further declare that to the best of my knowledge and belief the following is a true and correct translation made by me of Japanese Patent Application No. 2000-190891 filed on June 26, 2000.

Date: January 25, 2006

Kyozo Omori

KYOZO OMORI

[DOCUMENT] Patent Application

[OUR REFERENCE NO.] 2022520267

[FILING DATE OF APPLICATION] June 26, 2000

[To] Commissioner, Patent Office

[INTERNATIONAL PATENT CLASSIFICATION] G06F 7/00

G11B 7/00

[INVENTOR]

[ADDRESS] c/o Matsushita Electric Industrial Co., Ltd.

1006, Kadoma, Kadoma-City, Osaka

[NAME] Tokuo NAKATANI

[APPLICANT]

[CODE NO.] 000005821

[NAME] Matsushita Electric Industrial Co., Ltd.

[PATENT AGENT]

[CODE NO.] 100097445

[PATENT ATTORNEY]

[NAME] Fumio IWAHASHI

[APPOINTED PATENT AGENT]

[CODE NO.] 100103355

[PATENT ATTORNEY]

[NAME] Tomoyasu SAKAGUCHI

[APPOINTED PATENT AGENT]

[CODE NO.] 100109667

[PATENT ATTORNEY]

[NAME] Hiroki NAITO

[CHARGES]

[RECEIPT NO.] 011305

[AMOUNT] ¥21,000

[LIST OF ENCLOSURES]

Specification 1

Drawings 1

Abstract 1

[POWER OF ATTORNEY NO.] 9809938

[DOCUMENT] Specification

[TITLE OF THE INVENTION] Recorder

[CLAIMS]

[CLAIM 1] A recorder for recording video data, which is generated by compressing and encoding a video signal, onto a recording medium, and at the same time, generating management information for managing the video data, the recorder comprising:

a recording unit operable to record the video data onto the recording medium; and

a nonvolatile memory, wherein

in preparation for an occurrence of a power failure during recording of the video data onto the recording medium, the nonvolatile memory stores therein:

the number of recorded PGs;

the number of VOBs corresponding to the recorded PGs on a one-to-one basis;

a first set of: a recording start time of a VOB, a pointer pointing to attribute information referred to by the VOB; a playback start time code of the VOB; and the number of VOBUs constituting the VOB, the first set corresponding, on a one-to-one basis, to the VOBs constituting each PG; and

a second set of: an offset of the last data of the first Intra-coded picture in a VOBU; a data size of the VOBU; and a playback time of the VOBU, the second set corresponding, on a one-to-one basis, to the VOBUs constituting each VOB and being generated on presumption that the start of the VOBU is "0".

[CLAIM 2] A recorder for recording video data, which is generated by compressing and encoding a video signal, onto a recording medium, and at the same time, generating management information for managing the video data, the recorder comprising:

a recording unit operable to record the video data onto the recording medium; and

a nonvolatile memory, wherein

in preparation for an occurrence of a power failure during recording of the video data onto a disc, the nonvolatile memory stores therein:

the management information;

the number of recorded PGs;

the number of VOBs corresponding to the recorded PGs on a one-to-one basis;

a first set of: a recording start time of a VOB, a pointer pointing to attribute information referred to by the VOB; a playback start time code of the VOB; and the number of VOBUs constituting the VOB, the first set corresponding, on a one-to-one basis, to the VOBs constituting each PG; and

a second set of: an offset of the last data of the first Intra-coded picture in a VOBU; a data size of the VOBU; and a playback time of the VOBU, the second set corresponding, on a one-to-one basis, to the VOBUs constituting each VOB and being generated on presumption that the start of the VOBU is "0".

[CLAIM 3] The recorder recited in CLAIM 1 or CLAIM 2, wherein

if the recorder was re-started after a power failure occurred, the management information for the video data, which was being

recorded when the power failure occurred, is restored using (i) a management information file recorded on the recording medium and (ii) information recorded on the nonvolatile memory.

[DETAILED DESCRIPTION OF THE INVENTION]

0001

[FIELD OF THE INVENTION]

The present invention relates to a technology for recording digital data of video onto a recording medium.

0002

[DESCRIPTION OF THE RELATED ART]

In the history of establishing the technology for recording video data, initially, tape mediums were mainly used as the recording mediums. As it became possible to record video data, a function to perform special playbacks (fastforward, rewinding and the like) was provided. Such a function has become commonplace nowadays. When data is recorded onto a tape medium, the video data is consecutively recorded onto the tape. As a result, the recording order of video data on the tape is used as the playback order as it is, and a special playback is achieved by performing an intermittent playback while physically fastforwarding or rewinding the tape.

0003

And then optical discs such as CD were developed and came to practical use as mediums for recording data thereon. Disc mediums are superior than the tape mediums in accessibility. When tape is used, it is necessary to move the tape to a point where desired data is recorded. For this purpose, the tape is moved one-dimensionally, which takes much time.

0004

However, in the case of disc mediums, a two-dimensional moving process is performed. That is to say, while the disc is moved, the pickup is moved to read data. Due to this construction, disc mediums have remarkably higher accessibility than the tape mediums. To fully make use of the accessibility, management information is required to manage where on the disc the data is recorded. When disc mediums first appeared, they did not attract much attention as the mediums for recording video data thereon due to their small capacity (among them, the video CD has been put into practical use, but not so popular). It is considered that the unpopularness of the medium is also attributable to the limitation "read-only".

0005

In recent years, however, DVD-RAM, which is a phase-change optical disc having several giga bytes of capacity, appeared. Coupled with the practical use of MPEG (MPEG2), which is an encoding standard for digital audio-visual data (hereinafter, AV data), the DVD-RAM is being used as an audio-visual recording/playback medium, as well as being used for computers. As the specifications of information required to achieve the special playbacks and the like on the video data recorded on the DVD-RAM, DVD Specifications for Rewritable/Re-recordable Discs were established and issued. This completes the technologies required for recording video data onto recording mediums.

0006

(Explanation of MPEG)

The AV data recorded on the DVD-RAM needs to conform to an international standard called MPEG (ISO/IEC13818).

0007

Although DVD-RAM has a large capacity of several giga bytes, it is not enough to record non-compressed digital AV data as it is. This necessitates the use of a method of compressing AV data and recording the compressed AV data. As the AV data compression method, MPEG (ISO/IEC13818) has become widespread. Owing to the recent improvement in the LSI technology, MPEG codec (expansion/compression LSI) has come to practical use. This has enabled DVD recorders to perform MPEG expansion/compression.

0008

MPEG has the following two main characteristics for realizing highly-efficient data compression. One characteristic is that its moving-image data compression adopts a compression method using the temporal correlation between frames, as well as a conventional compression method using the space frequency. For data compression, MPEG classifies each frame (also referred to as "picture" in MPEG) into three types: I-picture (Intra-coded picture); P-picture (Predictive picture --- a picture using the intra-coding and references from the past); B-picture (Bidirectionally-predictive picture --- a picture using the intra-coding and references from the past and future).

0009

Fig. 1 shows relationships among I-, P-, and B-pictures. As shown in Fig. 1, a P-picture refers to an I- or P-picture that is closest thereto in the past, and a B-picture refers to an I- or P-picture that is closest thereto in the past and future. Also, since, as shown in Fig. 1, a B-picture refers to an I- or P-picture in the future, there may occur a case where the display order of the pictures does

not match the coding order thereof.

0010

To realize, in a playback of data stored in a storage medium, trick plays such as fastforward, rewinding, and a playback from a middle position, MPEG defines a structure called GOP (Group Of Pictures). More specifically, a GOP is composed of a set of frames that includes at least one I-picture so that a random access can be performed in units of GOPs. This enables trick plays to be realized by performing a skip playback, playing back only the I-pictures in GOPs.

0011

The second characteristic of MPEG is that it can dynamically assign amounts of coding in units of pictures in accordance with the complexity level of the images. An MPEG decoder includes an input buffer. It is possible to assign a large amount of coding to a complex image that is difficult to compress, by preliminarily storing data in the decoder buffer.

0012

The audio data used in DVD-RAM can be selected from three types: MEPG audio using compressed data; Dolby digital (AC-3) using compressed data; and LPCM using non-compressed data. The Dolby digital and the LPCM uses a fixed bit rate. For the MEPG audio, it is possible to select a size from several sizes in units of audio frames, though it is not so large as a video stream.

0013

The AV data described above is multiplexed into one stream by a method called MPEG system. Fig. 2 shows the construction of an MPEG system. The element 21 is a pack header, 22 a packet header, and 23 a payload. The MPEG system has a hierarchical structure composed

of packs and packets. Each packet includes a packet header 22 and a payload 23. The AV data is divided into pieces having an appropriate size, and the divided pieces of AV data are stored into the payloads 23, respectively.

0014

Recorded in the packet header 22 are, as information of the AV data stored in the payload 23: stream ID for identifying the stored data; DTS (Decoding Time Stamp) indicating the decoding time of the data in the payload, in the accuracy of 90 kHz; and PTS (Presentation Time Stamp) indicating the presentation time of the data in the accuracy of 90 kHz (it should be noted here that the DTS is omitted when, as is the case with audio data, decoding and presentation are performed at the same time). Each pack is composed of a plurality of packets.

0015

In the case of the DVD-RAM, each packet is used as a pack. And accordingly, each pack is composed of a pack header 21 and a packet (composed of a packet header 22 and a payload 23). Recorded in the packet header 22 is SCR (System Clock Reference) that indicates the time at which the data in the pack is input into the decoder buffer, in the accuracy of 27 MHz. In DVD-RAM, the MPEG system stream as described above is recorded on presumption that one pack corresponds to one sector (= 2,048 bytes).

0016

The following describes the decoder for decoding the above-described MPEG system stream. Fig. 3 is a block diagram showing a decoder model (P-STD) of the MPEG system decoder. The element 31 is an STC (System Time Clock) for providing a standard time in the decoder, 32 is a demultiplexer for decoding, namely, demultiplexing

the system stream, 33 is a video buffer for a video decoder, 34 is the video decoder, 35 is a reorder buffer for temporarily storing I- and P-pictures to absorb the difference between the data order and the display order that occurs with the I-, P-, and B-pictures, 36 is a switch for adjusting the output order of the I-, P-, and B-pictures stored in the reorder buffer, 37 is an audio buffer for an audio decoder, and 38 is the audio decoder.

0017

The MPEG system decoder processes the MPEG system stream as follows. The demultiplexer 32 inputs a pack when the time provided by the STC 31 matches the SCR written in the pack header of the pack. The demultiplexer 32 decodes the stream ID in the packet header, transfers the payload data to the decode buffers for each stream, and extracts the PTS and DTS from the packet header. The video decoder 34 extracts the picture data from the video buffer 33 at a time when the time provided by the STC 31 matches the DTS, performs the decoding process, stores the I- and P-pictures into the reorder buffer 35, and outputs the B-pictures for display. When the video decoder 34 is decoding the I- and P-pictures, the switch 36 is set to output the I- or P-pictures to the reorder buffer 35, and when the video decoder 34 is decoding the B-pictures, the switch 36 is set toward the video decoder 34. The audio decoder 38, as is the case with the video decoder 34, extracts one audio frame of data from the audio buffer 37 at a time when the time provided by the STC 31 matches the PTS (there is no DTS for audio), and performs the decoding process.

0018

Next, the MPEG system stream multiplexing method will be described with reference to Fig. 4. In Fig. 4, (a) indicates video

frames, (b) a video buffer, (c) an MPEG system stream, and (d) audio data. The horizontal axis represents a time axis that is common to (a) through (d). In (b), the vertical axis represents the amount of buffer occupation (amount of data stored in the video buffer), and the thick line indicates the temporal transition of the amount of buffer occupation. The slant of the thick line corresponds to the video bit rate, indicating that the data is input into the buffer at a constant rate. Also, (b) shows that the amount of buffer occupation decreases at regular intervals. This indicate that the data is decoded. Further, the points at the intersections of slant dotted lines and the time axis indicate the times at which the data of the video frames starts to be transferred to the video buffer.

0019

Now, an explanation using a complex image A in the video data will be given. As shown in (b) of Fig. 4, since the image A requires a large amount of coding, the data transfer to the video buffer is started at time t1 prior to the decoding time of the image A. (The time period from the data input start time t1 to the decoding time is referred to as vbv_delay) For this reason, the data, as the AV data, starts to be multiplexed at the position (time) of the video pack indicated by the arrow with hatching. On the other hand, in the case of audio data that does not require the dynamic control of the amount of coding as video data, there is no need to start transferring data much earlier than the decoding time, and generally the audio data is multiplexed a little before the decoding time.

0020

Therefore, in terms of video data and audio data that are played back at the same time, the video data is started to be multiplexed

earlier than the audio data. It should be noted here that MPEG defines that the time period for which data can be stored in the buffer is limited, and that all the data except for still picture data should be output from the buffer to the decoder within one second after the data is input into the buffer. Accordingly, the difference between video data and audio data in multiplexing is one second at the maximum (more accurately, the time for reordering of the video data may be added to the difference).

0021

In the above-described example, video precedes audio. However, theologically, audio may precede video. Such data can be intentionally generated by, for example, preparing a simple image with high compression rate for the video data and unnecessarily transferring the audio data. It should be noted here that the time period allowed for the preceding is one second at the maximum due to the limitation by MPEG.

0022

(Logical Structure of DVD-RAM)

Here, the logical structure of DVD-RAM will be described. The portion (a) of Fig. 5 shows the data structure of the file system in the disc, and (b) shows the physical sector addresses in the disc.

0023

The lead-in area is provided at the start of the physical sector addresses. The lead-in area stores a standard signal necessary for stabilizing the servo, an identification signal against other mediums, and the like. The lead-in area is followed by the data area in which logically effective data is recorded. The physical sector addresses has the lead-out area at the end. The lead-out area stores,

as the lead-in area, a standard signal and the like.

0024

Recorded at the start of the data area is management information for the file system, the management information called volume information. As shown in (a) of Fig. 5, use of the file system enables data in the disc to be treated as a directory or a file.

0025

In the structure defined in the VIDEO RECORDING standard, as shown in (b) of Fig. 5, the management information is present under the DVD_RTAV directory that is immediately under the ROOT directory. Under the DVD_RTAV directory, there are a management information file "VR_MANGR.IFO" (hereinafter referred to as IFO file) and a plurality of (at least one) AV files. The AV file is classified into three files: moving image (VR_MOVIE.VRO); still image (VR_STILL.VRO); and audio dubbed for the still image (VR_AUDIO.VRO). One IFO file is provided as information for managing the three AV files.

0026

(Explanation of Management Information File in VIDEO RECORDING Standard)

Here, the structure of the IFO file defined in the VIDEO RECORDING standard will be explained. This explanation centers on the management information for video, and in particular centers on PG (explanation of the part that is irrelevant to the present patent application is omitted).

0027

As shown in Fig. 6, the IFO file includes three video-related tables: VOB_STI (VOB stream attribute information) table; VOBI (VOB information) table; and PGCI (PGC information) table. The VOB is an

MPEG program stream. The Cell is a logical playback unit that refers to a given partial section (or the whole section) in the VOB. The PGC defines the playback order of the Cell. In the VIDEO RECORDING standard, strictly speaking, the VOB differs from the MPEG system stream. However, in this explanation, they are treated as the same one (The following is the difference. The system stream should end with the program end code. In contrast, there is no such definition in regards with VOBs in the VIDEO RECORDING standard).

0028

A VOB is a set of VOBUs. The VOBU is a data unit that includes an MPEG program stream generated by multiplexing one or more sets of GOPs of MPEG video data, its program stream, and a plurality of interleaved audio packs. It should be noted here that the GOPs included in one VOBU are complete themselves in the VOBU. Also, the playback time period of one VOBU is limited to a certain range, and the encoder needs to generate the VOBU to meet the limitation.

0029

The aforesaid VOBI in the IFO file is the management information of the above-described VOB. The VOBI table records therein the number of VOBIs (VOB_SRP_Ns) and VOBIs. The VOBI includes the type of the VOB (VOB_Type), playback start time (VOB_Start_PTM), playback end time (VOB_End_PTM), information relating to the time at which the start of the VOB is recorded (VOB_REC_TM), a reference pointer to a VOB STI (to be described later) that shows the attribute information of the VOB (VOB_STIN), and time map information (TMAPI).

0030

The TMAPI is management information used for a special playback or a jump playback, and includes information regarding a VOBU that

constitutes a VOB. More specifically, the TMAPI includes TIME MAP GENERAL information (TMAP_GI), a time map entry (TM_ENT), and a VOBU entry (VOBU_ENT). As shown in the upper-left portion of Fig. 7, TMAP_GI includes a VOB address offset (ADE_OFS), a playback time offset from the start of the VOB of FIRST TM_ENT (TM_OFS), the number of time map entries (TM_ENT_Ns), and the number of VOBU entries (VOBU_ENT_Ns).

0031

Fig. 7 shows the TMAPI in relation to the stream. As the figure indicates, ADR_OFS is OFFSET from the stream file start to the VOB start on presumption that the stream file start is "0". TM_OFS is OFFSET from the playback start time of TM_ENT#1 on presumption that the VOB start time is "0". In general, TM_OFS is "0" immediately after recording. However, TM_OFS becomes a value other than "0" when, for example, the starting data of the VOB is deleted by editing. It should also be noted that the playback start time indicated by TM_ENT#j does not necessarily match the VOBU start time. This occurs since the VOBUs can have uneven playback time lengths. As a result, as shown in Fig. 7, TM_ENT has information called TM_DIFF as well as the VOBU number (VOBU_ENTN) to indicate a difference between the playback start time of the VOBU_ENT of the VOBU_ENTN and the playback start time of the TM_ENT itself. TM_ENT also has the start address (VOBU_ADR) of the VOBU it indicates. VOBU_ADR is an offset from the VOB start. By combining VOBU_ADR with ADR_OFS of TMAP_GI, it is possible to calculate an address from the start of the VR_MOVIE.VRO file, which makes it possible to directly access a given VOBU.

0032

VOB STI is attribute information of VOB being MPEG program stream. The information of VOB STI may be incorporated into each VOBI,

but the standard provides the specification such that VOBs having a common attribute refer to the same VOB STI, thereby restricting the size of the IFO file.

0033

As shown in Fig. 6, the above-mentioned VOB STI table records therein the number of VOB STIs (VOB_STI_Ns) and VOB STIs. Each VOB STI includes the Video Attribute (video attribute information), the number of audio streams (Number of Audio Streams), the number of sub picture streams (Number of Sub Picture Streams), Audio Attribute (audio attribute information), Sub Picture Attribute (sub picture attribute information), and Sub Picture Color Pallet.

0034

The PGCI table, which is management information of PGC, records therein the number of PGCIs (PG_Ns) and the PGCIs. Each PGCI includes the number of CellIs that are present in the PGC (C_Ns), and includes CellIs that are management information of each Cell.

0035

Each CellI includes a search pointer to the VOBI of VOB corresponding to the Cell (VOBI_SRP), Cell playback start time (Cell_Start_PTM), Cell playback end time (Cell_End_PTM), the number of entry points for the Cell (EPI_Ns), and entry point information (Cell_EPI) table.

0036

[THE PROBLEMS THE INVENTION IS GOING TO SOLVE]

As described in the Description of the Related Art, video recorders have been widely used for recording video data. The video recorders do not require special information for managing the data since the tape itself can be information indicating the positions

of the video data. Recent video recorders are ready for management information such as VISS (the information used to locate the start of a desired portion) attached to the video data. Such information is recorded together with the video data. As a result, if a power failure occurs during recording, any special recovery process is not required because both the video data and the management information have been recorded on the tape when the power failure occurs.

0037

However, in the case of devices (such as DVD recorders) that use disc mediums, management information for managing data addresses or the like is required. The reason for this is as follows. When data is recorded onto a disc medium, free spaces are detected, and the data is recorded into the free spaces. With such a construction, data may be recorded in a discontinuous manner (in the case where tape is used as the medium, the continuous recording of data is ensured by the physical property of the tape). Therefore, the recorded data needs to be managed with management information that indicates locations of the recorded data. The management information is created on the system memory while the stream is recorded, and when the recording is completed, the management information is written onto the disc.

0038

With such a construction, if a power failure occurs during recording, a mismatch occurs on the disc between the stream file and the management information.

0039

Any special recovery process would not be required if, as is the case with video recorders, the stream data and the management information are recorded onto the disc such that they are always

coordinated with each other. However, for the stream data and the management information to always be coordinated with each other on the disc, the management information should be written onto the disc together with the stream data with the same timing, each time the stream data is recorded. This is not a realistic way since writing onto the disc frequently occurs.

0040

Using the SRAM as the system memory would make it easy to perform the coordination process between the stream and the management information on the disc since management information on the SRAM is maintained even if a power failure occurs. However, use of the SRAM is prevented as far as possible since the SRAM is more expensive than the DRAM. When the SRAM is used, it is preferable that its size is as small as possible.

0041

[MEANS TO SOLVE THE PROBLEMS]

To solve the above problems, CLAIM 1 provides a recorder for recording video data, which is generated by compressing and encoding a video signal, onto a recording medium, and at the same time, generating management information for managing the video data, the recorder comprising: a recording unit operable to record the video data onto the recording medium; and a nonvolatile memory, wherein in preparation for an occurrence of a power failure during recording of the video data onto the recording medium, the nonvolatile memory stores therein: the number of recorded PGs; the number of VOBs corresponding to the recorded PGs on a one-to-one basis; a first set of: a recording start time of a VOB, a pointer pointing to attribute information referred to by the VOB; a playback start time code of the VOB; and the number

of VOBUs constituting the VOB, the first set corresponding, on a one-to-one basis, to the VOBs constituting each PG; and a second set of: an offset of the last data of the first Intra-coded picture in a VOBU; a data size of the VOBU; and a playback time of the VOBU, the second set corresponding, on a one-to-one basis, to the VOBUs constituting each VOB and being generated on presumption that the start of the VOBU is "0".

0042

CLAIM 2 provides a recorder for recording video data, which is generated by compressing and encoding a video signal, onto a recording medium, and at the same time, generating management information for managing the video data, the recorder comprising: a recording unit operable to record the video data onto the recording medium; and

a nonvolatile memory, wherein in preparation for an occurrence of a power failure during recording of the video data onto a disc, the nonvolatile memory stores therein: the management information; the number of recorded PGs; the number of VOBs corresponding to the recorded PGs on a one-to-one basis; a first set of: a recording start time of a VOB, a pointer pointing to attribute information referred to by the VOB; a playback start time code of the VOB; and the number of VOBUs constituting the VOB, the first set corresponding, on a one-to-one basis, to the VOBs constituting each PG; and a second set of: an offset of the last data of the first Intra-coded picture in a VOBU; a data size of the VOBU; and a playback time of the VOBU, the second set corresponding, on a one-to-one basis, to the VOBUs constituting each VOB and being generated on presumption that the start of the VOBU is "0".

0043

CLAIM 3 provides a recorder, wherein if the recorder was re-started after a power failure occurred, the management information for the video data, which was being recorded when the power failure occurred, is restored using (i) a management information file recorded on the recording medium and (ii) information recorded on the nonvolatile memory.

0044

[EMBODIMENTS OF THE INVENTION]

The present invention will be described in detail through a DVD recorder which is the first embodiment of the present invention. First, the basic structure of the DVD recorder will be explained.

0045

(Block Diagram of DVD Recorder)

Fig. 8 is a block diagram showing the DVD recorder. In Fig. 8, the element 81 is a user interface unit for displaying for a user and receiving requests from the user, 82 is a system control unit for managing and controlling the whole DVD recorder and for generating stream management information, 83 is an input unit composed of a camera and a microphone or a TV tuner, 84 is an encoder unit composed of a video encoder VE, an audio encoder AE, and a system encoder SE, 85 is an output unit composed of a monitor and a speaker, 86 is a decoder unit composed of a system decoder, an audio decoder, and a video decoder, 87 is a truck buffer, 88 is a drive, and 89 is a time management unit for managing the time in the system.

0046

(Normal Recording Operation)

First, the recording operation of the DVD recorder will be explained with reference to Fig. 8. The user interface unit 81 receives

a request from a user. The user interface unit 81 then conveys the user's request to the system control unit 82. The system control unit 82 analyzes the user's request and requests processes to each module. If the user's request is to record the video and audio, the system control unit 82 sets the encoder unit 84 to the settings requested by the user interface unit 81 (for example, how to compress video, or the system bit rate), generates forms of the VOB STI, VOBI, and CellI of the management information shown in Fig. 6, and requests the encoder unit 84 to encode the video frames and audio. In doing this, the system control unit 82 acquires the current time from the time management unit 89, and sets the acquired current time to VOB_REC_TM in VOBI.

0047

The encoder unit 84 generates video data by video-encoding video frames sent from the input unit 83, and at the same time generates audio data by audio-encoding audio sent from the input unit 83. The generated video and audio data are system-encoded and formed into a system stream that is an MPEG program stream. The generated system stream is transmitted to the truck buffer 87. At the same time, each time the system-encoding of a VOBU is completed, the encoder unit 84 notifies the system control unit 82 of VOBU information concerning the VOBU that was completely encoded. The system control unit 82 updates the management information shown in Fig. 6 based on the VOBU information.

0048

The VOBU information is classified into the following:

- VOBU Start PTM (video frame playback start time in VOBU);
- Reference Picture Size (size of the first I-picture when the VOBU

start is "0");

- VOBU Size (the number of multiplexed units);
- VOBU PB Time (playback time);
- Aspect ratio;
- AUDIO mode; and
- The number of AUDIO streams.

More specifically, the following processes are executed based on the above-listed information: update of TMAPI (addition of TMAP_ENT, VOBU_ENT); and update of VOB_End_PTM, Cell_End_PTM. The VOBU information that is received first after starting a recording is used to set VOB_Start_PTM, Cell_Start_PTM. After a predetermined amount of system stream is stored in the truck buffer 87, the system control unit 82 records the system stream data stored in the truck buffer 87 to a DVD-RAM disc via the drive 88.

0049

The stop request from the user is conveyed to the system control unit 82 via the user interface unit 81. The system control unit 82 then sends a video/audio recording stop instruction to the encoder unit 84. The encoder unit 84 ends the encoding with the system-encoding of the audio frame that was generated immediately after it, transfers the generated system stream data to the truck buffer 87, and conveys the end of the encoding process to the system control unit 82. The system control unit 82 records all the remaining system stream data stored in the truck buffer 87 onto the DVD-RAM disc via the drive 88.

0050

After the above-described operation, the system control unit 82 records the aforesaid VOBI and CellI onto the DVD-RAM disc via

the drive 88.

0051

Here, a recording pause operation of the recorder will be briefly explained, which is believed to help understand the later explanation. Upon receiving a notification of a pause request from the system control unit 82, the encoder unit 84 enters a pause state after emitting all the VOB data, which it currently encodes, into the truck buffer 87. This enables the encoder unit 84 to start encoding a new VOB immediately after the pause is released.

0052

If the encoder unit 84 enters a pause state keeping the VOB data it currently encodes, a communication between the system control unit 82 and the encoder unit 84 is required after the pause is released because up to which portion the VOB data has been encoded should be notified.

0053

The present embodiment adopts a method in which the encoder unit 84 enters a pause state after emitting all the VOB data, for the sake of simplicity.

0054

(Generating Data Table for Power Failure)

As explained earlier, after a recording is completed, the encoder unit 84 notifies the system control unit 82 of the VOBU information. Upon receiving the VOBU information, the system control unit 82 updates the VOBU map of the IFO file based on the received VOBU information, and at the same time, saves necessary data onto the SRAM. Fig. 9 shows the operation.

0055

Fig. 10 shows what information is actually saved to the SRAM. First, in regards with the number of PGs, the number of PGs increases each time a recording is performed. This is because the recorder shown in Fig. 8 records in units of PGs in each recording. Next, in regards with the number of Cells in the PG table of the SRAM and the number of VOBs, they are updated each time a VOB is generated or divided. More specifically, the number of Cells and the number of VOBs are updated when the system control unit 82 notifies the encoder unit 84 of a start of a recording or a release of a pause. Since the VOBs correspond to Cells on a one-to-one basis, the number of VOBs and the number of Cells in the PG table are updated at the same time. The number of VOBU_ENTs is initialized when the VOB table newly starts to be used.

0056

Next, in regards with VOB_REC_TM, M_VOB_STI_N, and VOB_Start_PTM in the VOB table, these information are set based on the VOBU information that is obtained first after the system control unit 82 requests a recording start or a pause release to the encoder unit 84. When the VOBU information is notified, the current time is acquired from the time management unit 89, and the acquired current time is set to VOB_REC_TM.

0057

The VOB_STI table in the IFO file is searched for the attribute information (Audio mode, Aspect ratio, the number of Audio streams) notified by the VOBU information. If a VOB_STI having the same attribute is detected, the table number is recorded in M_VOB_STI_N. If a target attribute is not found in the VOB_STI table, a new VOB_STI is generated.

0058

In this case, although it is not shown in Fig. 10, the whole VOB_STI is saved into the SRAM. This is because if a power failure occurs, VOB_STI can be recovered. As VOB_Start_PTM, VOB Start PTM in the VOBU information that is obtained first is written as it is. Lastly, the number of VOBU_ENTs is updated, namely increased by "1". The number of VOBU_ENTs is updated by "1" each time the VOBU information is notified.

0059

Next, in regards with the VOBU table, Reference Picture Size, VOBU Size, and VOBU PB Time among the notified VOBU information are saved into the SRAM. The table is updated, namely, increased each time the VOBU information is notified.

0060

In this example, the VOB table and the VOBU table are managed separately. However, the VOBU table may be incorporated into the VOB table such that a VOBU table is provided for each VOB.

0061

It may appear that only a small amount of information is saved into the SRAM compared with the amount of information of the IFO file shown in Fig. 6. However, other information necessary for the playback operation can be constructed from the information shown in Fig. 10.

0062

For example, Cell_Start_PTM needs not be saved since it is identical with VOB_Start_PTM. Cell_End_PTM and VOB_End_PTM can be calculated from VOB_Start_PTM and the total playback time of the VOBU. The TM_ENT information is not saved for the following reason. Even during recording, TM_ENT information is generated from the VOBU

information. As understood from this, TM_ENT table can be re-created from the VOBU information without TM_ENT information. Therefore, the TM_ENT information is not saved.

0063

The SRAM table information shown in Fig. 10 is cleared each time the IFO file is updated on the disc.

0064

It should be noted here that in the case of a system that is provided with an SRAM having a large amount of capacity, there may be no need to restrict the information to be saved into the SRAM, but the whole IFO file may be saved into the SRAM, as stated in "THE PROBLEMS THE INVENTION IS GOING TO SOLVE".

0065

(Power Failure Recovery Process)

Now, a power failure recovery process, which is performed after the system is re-started after a power failure occurs during recording, will be described.

0066

The VR_MOVIE.VRO file is updated in sequence on the disc during recording (the file system management information is not updated, thus a power failure recovery process is also required for the file system management information). As a result, on the disc after the re-start, the IFO file has not been changed from the state before the recording started, and the VR_MOVIE.VRO file has been changed up to the point when the power failure occurred. Here, an easier way to perform the recovery process would be to return the VR_MOVIE.VRO file to the state before the recording started, in coordination with the IFO file on the disc. However, the user would desire that the

content having been recorded before the power failure is restored as much as possible. Such conditions taken into account, the content having been recorded before the power failure is restored as much as possible (up to several seconds before the power failure occurred) based on the IFO file, the VR_MOVIE.VRO file, and the information on the SRAM shown in Fig. 10 on the disc.

0067

With reference to Fig. 11, first, in step 1101, fs_stream_size, namely, the file size of the VR_MOVIE.VRO file is obtained from the file system. Next, in step 1102, ifo_stream_size, namely, the logical file size is calculated from the IFO file. Then in step 1103, "pNo", a counter for PG, is initialized to "1", and "vobu_base", a variable for managing "OFFSET" that is used as the base of searching the vobu table, is initialized to "0". In step 1104, the initialized pNo is compared with the number of PGs in the SRAM information. If the number of PGs in the SRAM is smaller than pNo (namely, the number of PGs is 0), the process ends since it is recognized that this is not the case of a power failure during recording, and thus the power failure recovery is not necessary. In Fig. 11, when this is the case, steps 1111 and 1112 are executed. However, since fs_stream_size is equal to ifo_stream_size, there is no stream file portion to be deleted.

0068

Here, if the number of PGs is larger than pNo, meaning that a power failure occurred during a recording, the control moves to step 1105, and in this step, the number of Cells in the PG table in the SRAM is checked. If the number of Cells is 0, it means that although the recording request was sent to the encoder unit 84, the encoder unit 84 notified no VOBU information before the power failure occurred.

In this case, in step 1111, ifo_stream_size is subtracted from fs_stream_size to obtain del_size which is the size of the newly recorded stream data. Then, the data of del_size is deleted from the end of the VR_MOVIE.VRO file. If the number of Cells is 1 or more, the control moves to step 1106 in which new PGCI is created in the IFO file, and C_Ns is initialized to 0. Since new PGCI was created in this step, PGCI_Ns in the PGCI table is increased by "1" in step 1107.

0069

In step 1108, CellI (VOBI) included in the PG is restored. The operation in step 1108 will be described in detail later with reference to Fig. 12. After step 1108, in step 1109, the value of C_Ns in the PGCI is checked. If C_Ns is 0, meaning that there is no effective Cell (VOB) and that it is a vacant PG, the control moves to step 1113 in which the newly created PGCI is deleted, and in step 1114, PGCI_Ns is decreased by "1". The control then goes to steps 1111 and 1112 in which data is deleted from the VR_MOVIE.VRO file such that the size of the VR_MOVIE.VRO file becomes equal to the stream size calculated from the IFO file, and the process ends.

0070

If, in step 1109, C_Ns is judged to be 1 or more, meaning that the current PG is effective, the control moves to step 1110 to increase pNo by "1" and repeat the steps starting with step 1104 to restore the next PG. If it is judged in step 1104 that the updated pNo is larger than the number of PGs in the SRAM, it indicates that all the PGs to be restored in the power failure recovery process have been restored. Accordingly, the control moves to steps 1111 and 1112 in which data is deleted from the VR_MOVIE.VRO file such that the size of the VR_MOVIE.VRO file becomes equal to the stream size calculated

from the IFO file, and the process ends.

0071

Next, the CellI (VOBI) recovery process will be explained with reference to Fig. 12. First, in step 1201, "cNo", a counter for Cell, is initialized to "1". Then, in step 1202, cNo is compared with the number of Cells in the PG table in the SRAM. If cNo is larger than the number of Cells, it means that the PG includes no more Cell to be restored, and thus the process ends and returns to step 1109 in Fig. 11. If the number of Cells in the PG table in the SRAM is larger than cNo, it means that the PG includes a Cell to be restored. In this case, the control moves to step 1203 in which new CellI is created in the IFO file.

0072

In step 1204, C_Ns in PGCI is updated. Since Cells and VOBs correspond to each other on a one-to-one basis, in the next step 1205, new VOBI is created, TM_ENT_Ns and VOBU_ENT_Ns in VOBI are initialized to "0". And in step 1206, the number of VOBs (VOBs) in the IFO file is updated, namely, increased by "1". In step 1207, VOB_SRP_N in the CellI created in step 1203 is set to the number of VOBs updated in step 1206.

0073

In step 1208, the VOBU table is restored. This process will be described later with reference to Fig. 13. In step 1209 after step 1208, if it is judged that VOBU_ENT_Ns in VOBI is "0", meaning that the CellI and VOBI created in steps 1203 and 1205 do not include effective data, the control moves to step 1211 in which CellI is deleted, and then to step 1212 in which PGCI.C_Ns is decreased by "1".

0074

Further, in step 1213, the newly created VOBI is deleted, and then in step 1214, the number of VOBs is decreased by "1". The VOBI recovery process ends with this, and the control goes to step 1109 shown in Fig. 11. If, in step 1209, it is judged that VOBI.VOBU_ENT_Ns is "1" or more, meaning that the VOBI includes effective data, the control moves to step 1210 to increase cNo by "1" and repeat the steps starting with step 1202 to restore the next Cell.

0075

Now, the VOBU table recovery process will be explained with reference to Fig. 13. First, in step 1301, "vobuNo", a counter for vobu table, is initialized to "1", and "pb_time", a temporary variable for VOB playback time is initialized to "0".

0076

In step 1302, the number of VOBU_ENTs in the VOB table in the SRAM is compared with vobuNo. If vobuNo is larger than the number of VOBU_ENTs, meaning that all the VOBUs registered with the VOB have been restored, the control goes to step 1313 in which to restore the next VOB, a result of subtracting "1" from vobuNo is added to "vobu_base" that is the offset of the vobu table. If, in step 1302, it is judged that vobuNo is smaller than the number of VOBU_ENTs, meaning that the SRAM stores VOBU information to be restored, the control goes to step 1303.

0077

In step 1303, the stream size of the VR_MOVIE.VRO file and the stream size calculated from the IFO file are checked. Specific description of the step is shown in Fig. 14.

0078

Here, Fig. 14 is explained. First, in step 1401, the VOBU information that is the next candidate for being restored is obtained from the SRAM. Then, Tmp_size is obtained by adding ifo_stream_size, which is the logical stream size calculated from the current IFO file, to VOBU SIZE that is written in the VOBU information obtained from the SRAM. In step 1402, fs_stream_size obtained from the file system is compared with Tmp_size, and if Tmp_size is larger than fs_stream_size, "TRUE" is returned, and if Tmp_size is equal to or smaller than fs_stream_size, "FALSE" is returned.

0079

If TRUE is returned in step 1304 shown in Fig. 13, it means that all the VOBU information saved in the SRAM has been restored, and the VOBI recovery process ends. If FALSE is returned in step 1304, it means that the VOBU checked in Fig. 14 needs to be registered with the IFO file, and the control moves to step 1305 to create a new VOBU_ENT and increase VOBU_ENT_Ns in VOBI by "1". Also, the information in the new VOBU_ENT is set based on the information of VOBU table in the SRAM (Reference Picture Size, VOBU PB Time, VOBU Size).

0080

In step 1307, PB Time of the VOBU, on which attention is focused currently, is added to the temporary variable pb_time. Then, in step 1308, to check whether TM_ENT needs to be added, the pb_time is compared with a result of multiplying VOBI.TM_ENT_Ns by TIME UNIT length defined in the standard (frame unit, 10 seconds, and in this example, the length is "600" when NTSC is taken into consideration). If pb_time is larger than the result, a new TM_ENT needs to be created, and in step 1309, a new TM_ENT is created, and VOBI.TM_ENT_Ns is increased by "1".

0081

In step 1310, the information of the newly added TM_ENT is set. More specifically, the following settings are performed.

TM#ENT.VOBU#ENTN = vobuNo

TM#ENT.ADR = ifo#stream#size (a value calculated at the current point in time) - VOBU table in SRAM [vobuNo].VOBU Size

TM#ENT.DIFF = 600 * (VOBI.TM#ENT#Ns - 1) - (pb#time - VOBU table in SRAM [vobuNo + vobu#base] PBTime)

0082

After VOBU_ENT and TM_ENT are set, CellI and VOBI are set in step 1311. More specifically, VOB_End_PTM and CELL_End_PTM are updated. These values are updated by performing calculations based on the following expressions.

0083

In the case of NTSC:

VOB#END#PTM (new) = VOB#END#PTM (old) + (VOBU table in SRAM [vobuNo + vobu#base] PBTime/2) * 3003

CELL#END#PTM (new) = CELL#END#PTM (old) + (VOBU table in SRAM [vobuNo + vobu#base] PBTime/2) * 3003

In the case of PAL:

VOB#END#PTM (new) = VOB#END#PTM (old) + (VOBU table in SRAM [vobuNo + vobu#base] PBTime/2) * 3600

CELL#END#PTM (new) = CELL#END#PTM (old) + (VOBU table in SRAM [vobuNo + vobu#base] PBTime/2) * 3600

In the above expressions, the VOBU table in SRAM [vobuNo + vobu#base] PBTime is divided by “2”. This is because the PB Time is represented in units of fields.

0084

After step 1311 is executed, the control moves to step 1312 to update vobuNo, and then the control returns to step 1302 to repeat the steps therefrom. In Fig. 13, vobu_base is added to vobuNo when the information of the VOBU table in the SRAM is obtained. The reason for this is as follows. VOBUs of all VOBs are saved in one VOBU table, and vobu_base indicates the total number of VOBUs up to the VOB that is immediately before the current one. And therefore, the VOBU information of the current VOB can be obtained by searching the VOBU table using the vobu_base as the base.

0085

[EFFECTS OF THE INVENTION]

The present invention can restore the program that was being recorded when a power failure occurred, and in doing this, can use only half or less the size of the SRAM, compared with the case where the whole IFO is stored in the SRAM, which is more expensive than the DRAM, when the SRAM is used for the power failure recovery (more specifically, the present invention can use only 200KB or less of the SRAM space according to the current VIDEO RECORDING standard).

[BRIEF DESCRIPTION OF THE DRAWINGS]

Fig. 1 shows relationships among pictures in the MPEG video stream.

Fig. 2 shows the construction of the MPEG system stream.

Fig. 3 is a block diagram showing the construction of the MPEG system decoder (P-STD).

Fig. 4:

(a) a figure showing video data;

(b) a figure showing video buffer;

(c) a figure showing MPEG system stream; and

(d) a figure showing audio data.

Fig. 5:

(a) a figure showing directory structure; and

(b) a figure showing a physical arrangement on the disc.

Fig. 6 shows management information data.

Fig. 7 shows relationships between the stream and the TIME MAP information.

Fig. 8 shows the construction of the DVD recorder.

Fig. 9 shows processing the stream data and the VOBU information when the VOBU is notified.

Fig. 10 shows what information is saved into the SRAM.

Fig. 11 is a flowchart of the power failure recovery process (PGCI information).

Fig. 12 is a flowchart of the power failure recovery process (CellI/VOBI information).

Fig. 13 is a flowchart of the power failure recovery process (time map information).

Fig. 14 is a flowchart of the power failure recovery process (checking coordination between the VR_MOVIE.VRO file and the IFO file).

[DESCRIPTION OF CHARACTERS]

21 pack header

22 packet header

23 payload

31 STC

32 demultiplexer

33 video buffer

34 video decoder

35 reorder buffer

36 switch

37 audio buffer

38 audio decoder

81 user interface unit

82 system control unit

83 input unit

84 encoder unit

85 output unit

86 decoder unit

87 truck buffer

88 drive

89 time management unit

[DOCUMENT] Abstract

[SUMMARY]

[AIM] To reduce the SRAM space used for recovery from the power failure that occurs during recording.

[MEANS TO ACHIEVE THE AIM] In preparation for an occurrence of a power failure during recording, the following data is stored into the SRAM while the recording is performed: the number of recorded PGs; the number of Cells for each PG; for each VOB: VOB_REC_TIME; VOB STI search pointer; VOB_Start_PTM; and the number of VOBU_ENTs; and for each VOBU: Reference Picture Size, VOBU Size, and PB Time. In the power failure recovery process, the management information for the stream that was being recorded when the power failure occurred is constructed from the IFO file on the disc and the above-indicated pieces of data in the SRAM.

[SELECTED FIGURE] Fig. 10

FIG. 1

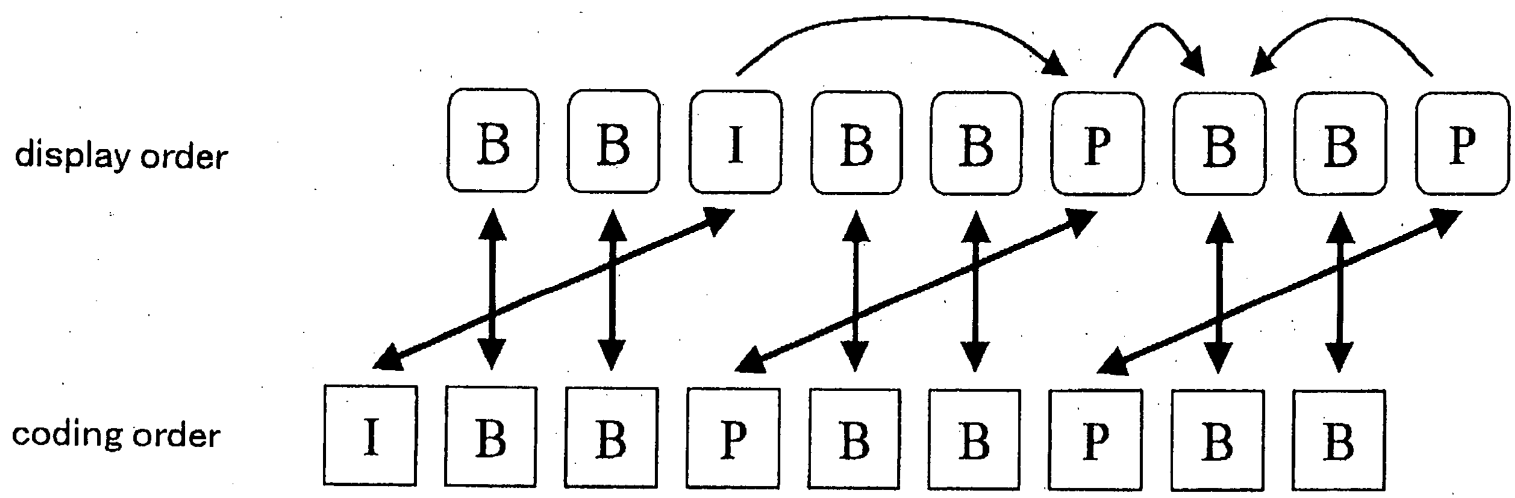
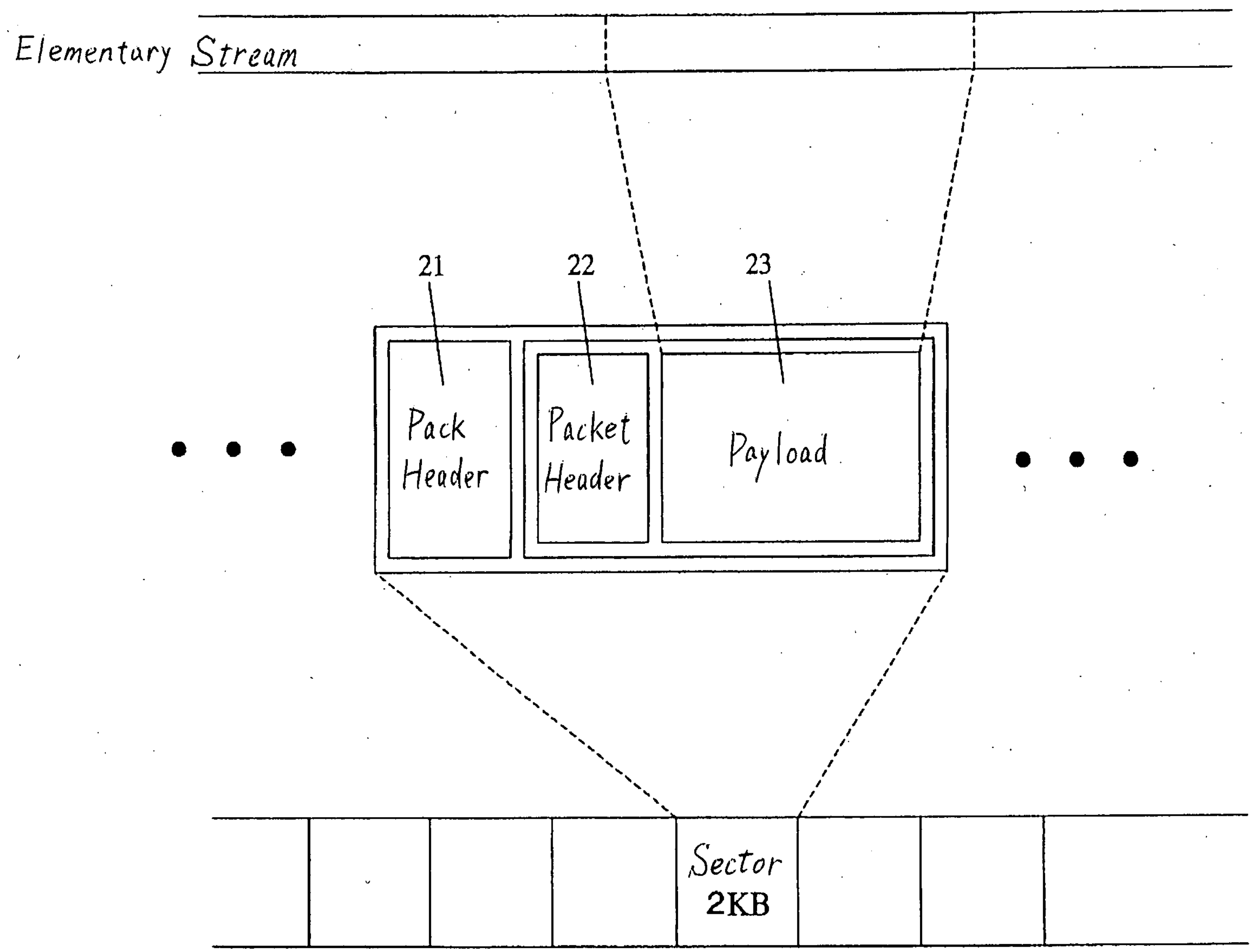
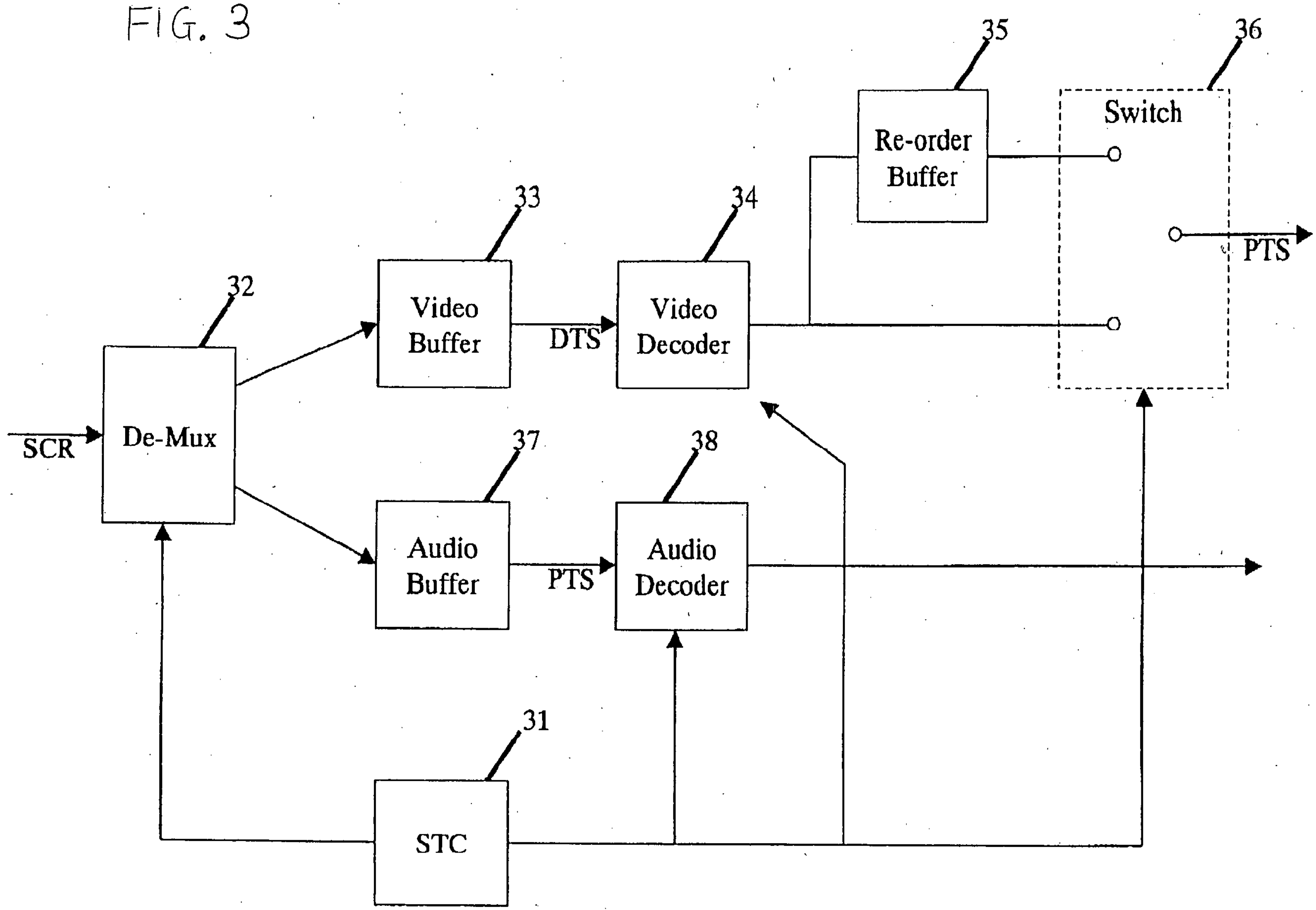


FIG. 2



Elementary Stream
21
22
23
Pack Header
Packet Header
Payload
Sector
2KB

FIG. 3



32
De-Mux
SCR
33
Video
Buffer
DTS
34
Video
Decoder
35
Re-order
Buffer
36
Switch
PTS
37
Audio
Buffer
PTS
38
Audio
Decoder
31
STC

FIG. 4

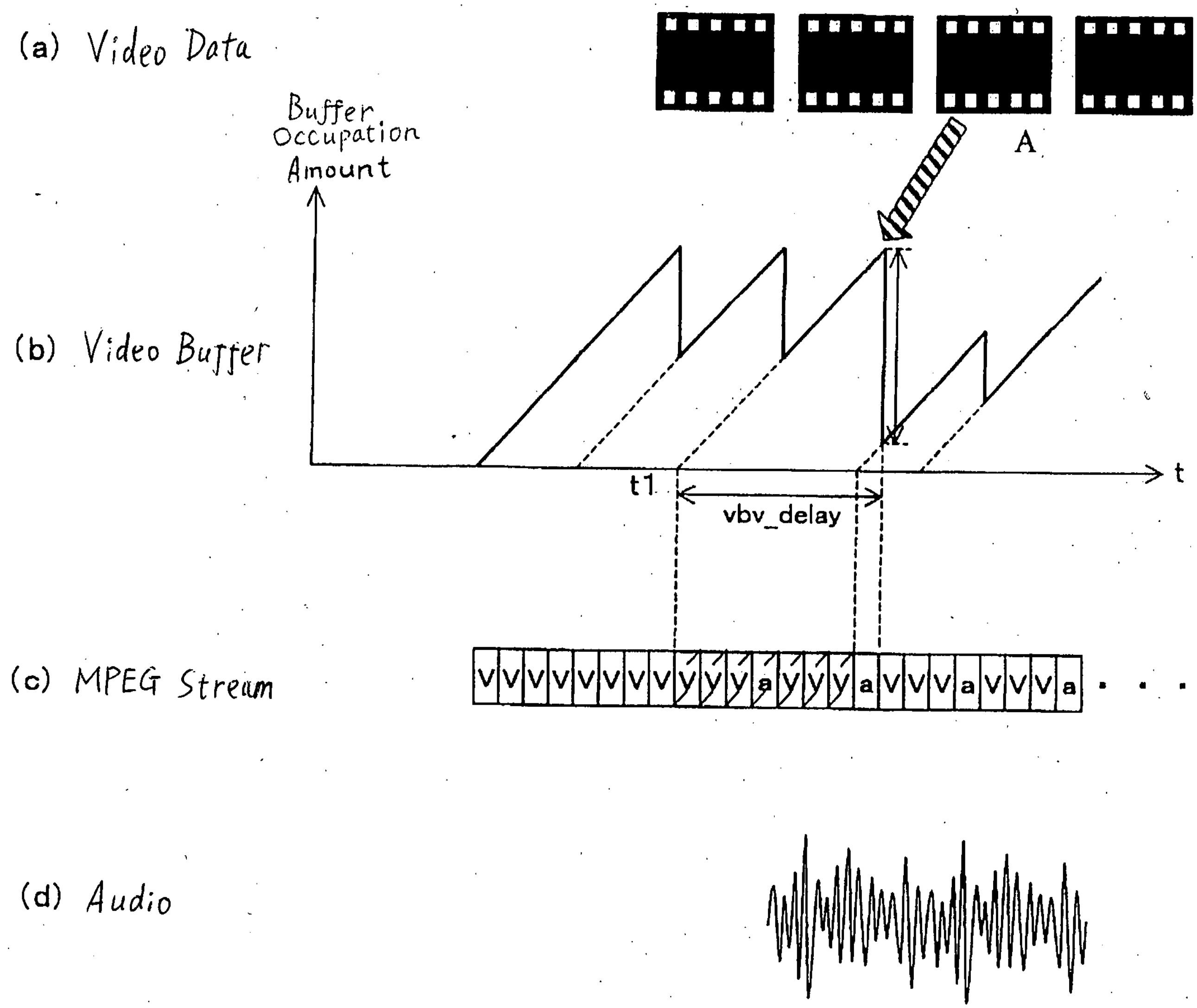

(a) Video Data
A
(b) Video Buffer
Buffer Occupation Amount
t
t1
vbv_delay
(c) MPEG Stream
v v v v v v v v v v v a v v v a v v v a v v v a
(d) Audio

FIG. 5

(a)

(b)

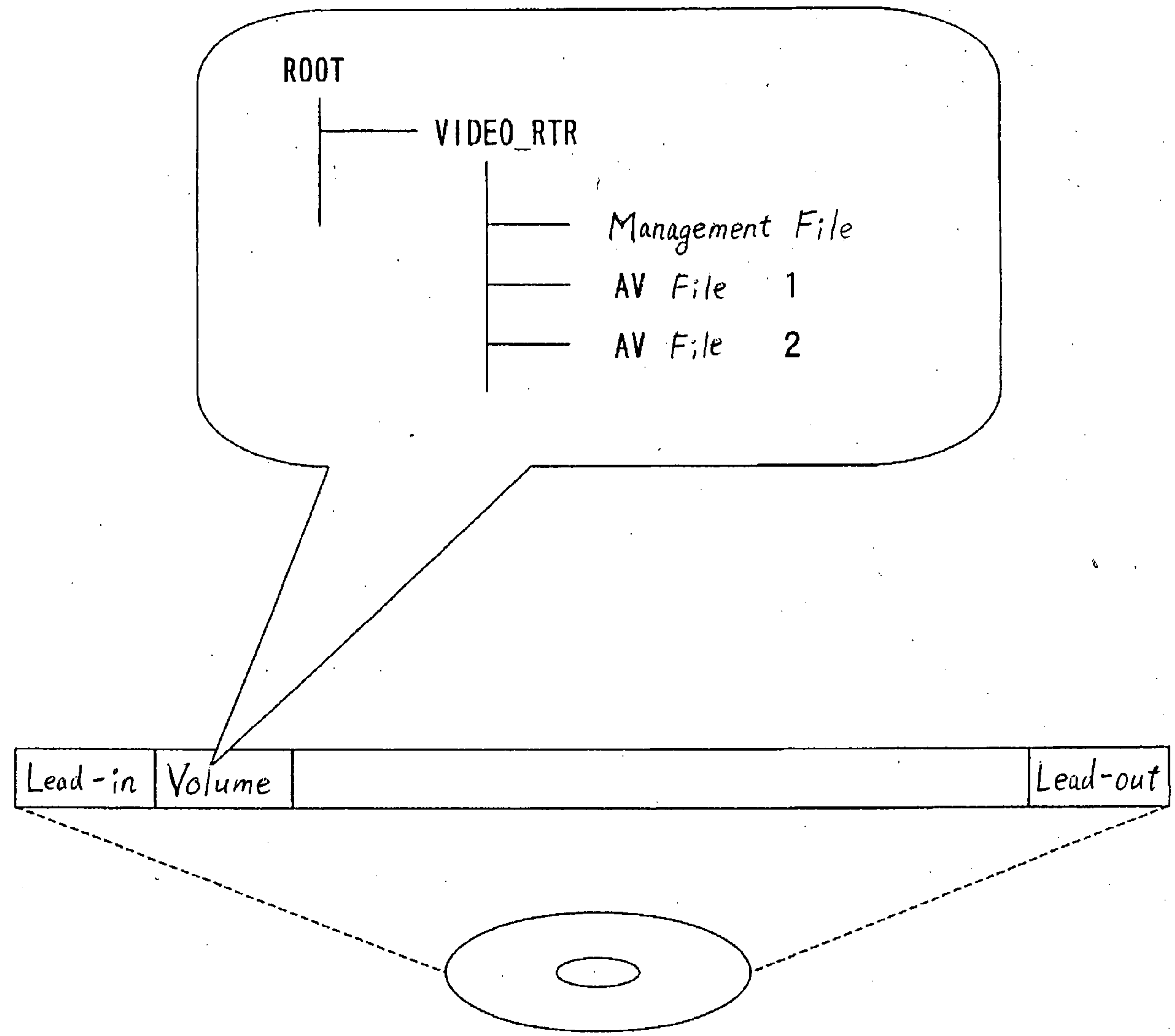
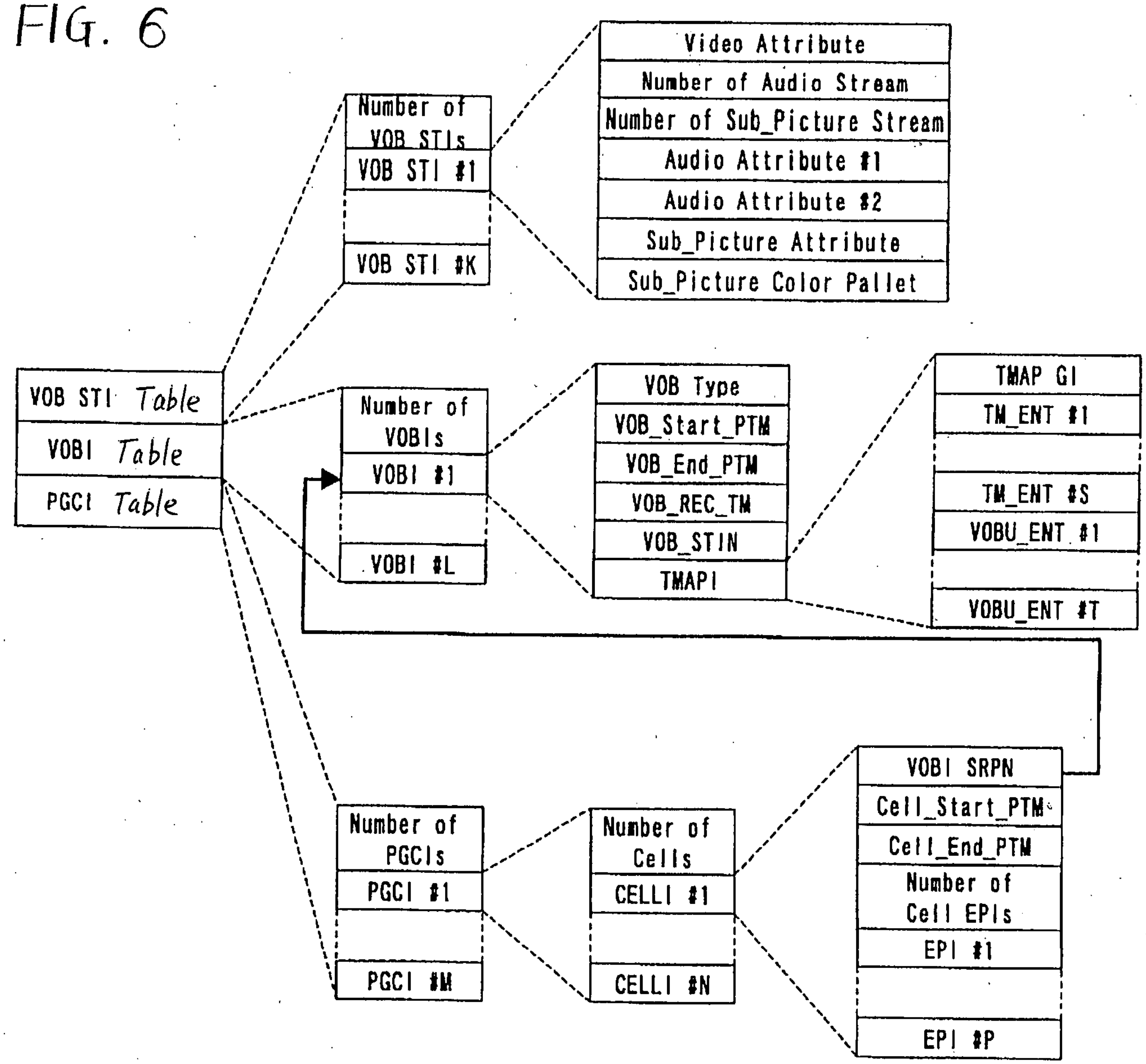
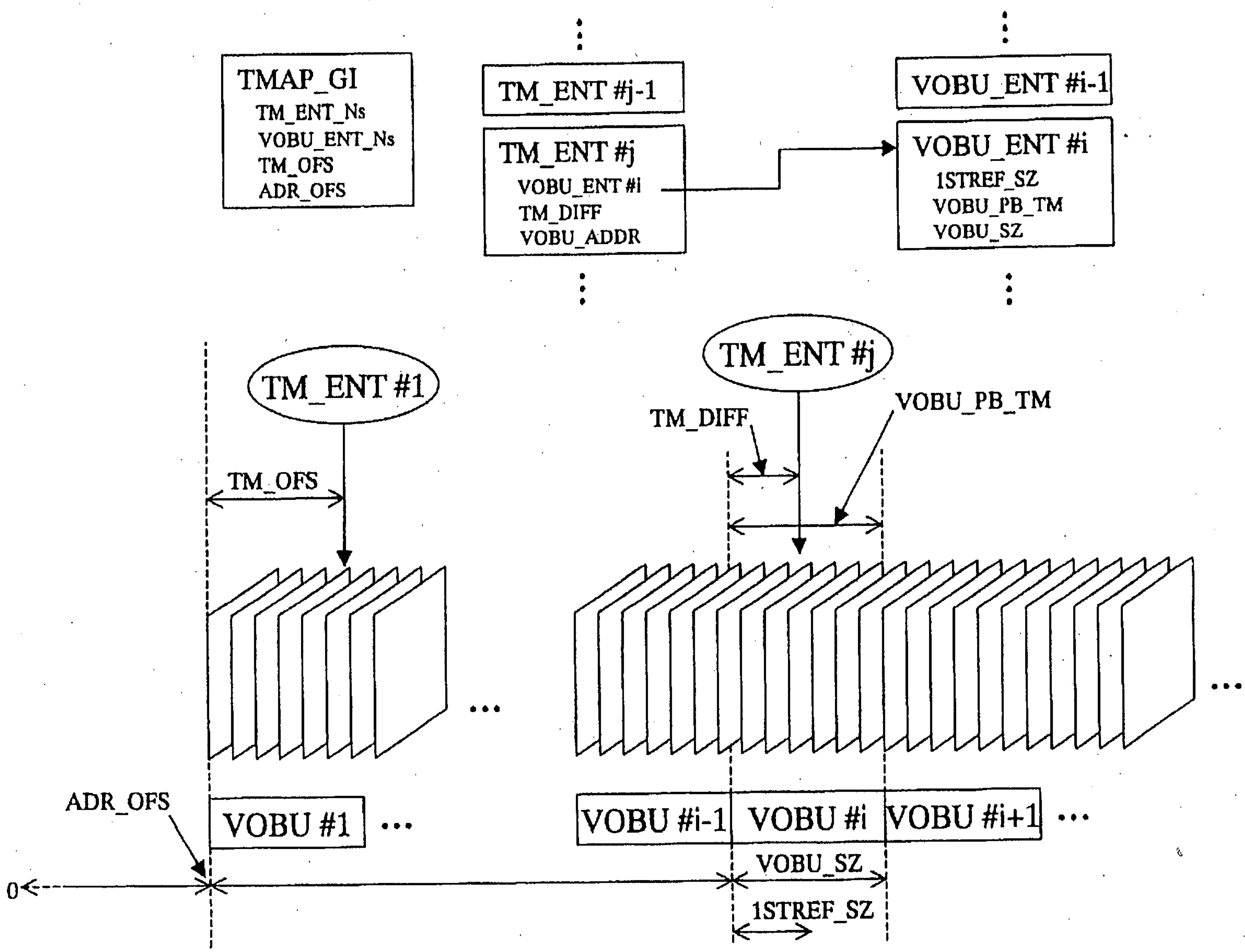

ROOT
VIDEO_RTR
Management File
AV File 1
AV File 2
Lead-in
Volume
Lead-out

FIG. 6



VOB STI Table
VOBI Table
PGCI Table
Number of VOB STIs
VOB STI #1
VOB STI #K
Video Attribute
Number of Audio Stream
Number of Sub_Picture Stream
Audio Attribute #1
Audio Attribute #2
Sub_Picture Attribute
Sub_Picture Color Pallet
Number of VOBIs
VOBI #1
VOBI #L
VOB Type
VOB_Start_PTM
VOB_End_PTM
VOB_REC_TM
VOB_STIN
TMAPI
TMAP GI
TM_ENT #1
TM_ENT #S
VOBU_ENT #1
VOBU_ENT #T
Number of PGCIs
PGCI #1
PGCI #M
Number of Cells
CELLI #1
CELLI #N
VOBI SRPN
Cell_Start_PTM
Cell_End_PTM
Number of Cell EPIs
EPI #1
EPI #P

FIG. 7



TMAP_GI
TM_ENT_Ns
VOBU_ENT_Ns
TM_OFS
ADR_OFS
TM_ENT #j-1
TM_ENT #j
VOBU_ENT #i
TM_DIFF
VOBU_ADDR
VOBU_ENT #i-1
VOBU_ENT #i
1STREF_SZ
VOBU_PB_TM
VOBU_SZ
TM_ENT #1
TM_OFS
TM_ENT #j
TM_DIFF
VOBU_PB_TM
ADR_OFS
VOBU #1
VOBU #i-1
VOBU #i
VOBU #i+1
VOBU_SZ
1STREF_SZ
0

FIG. 8

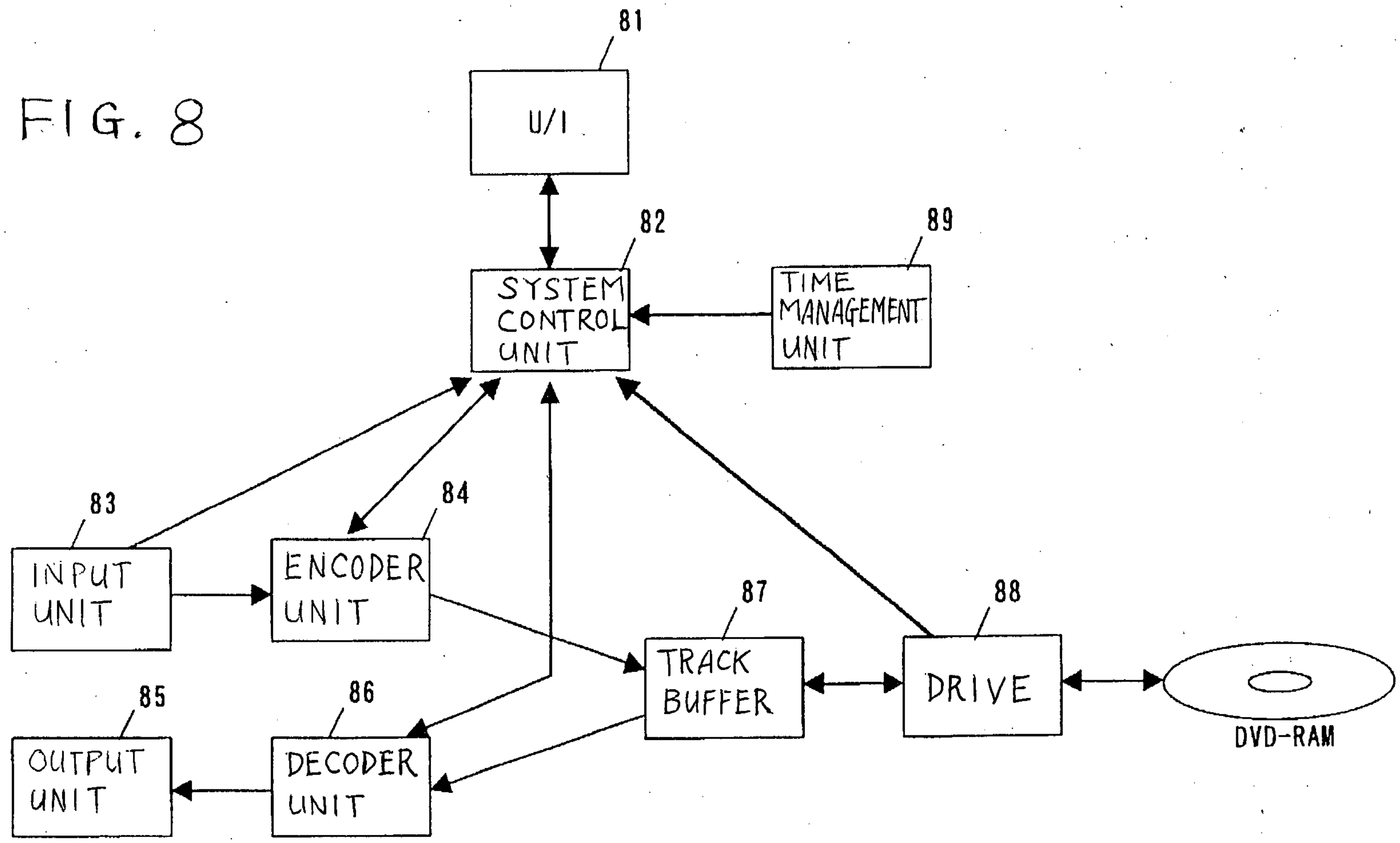
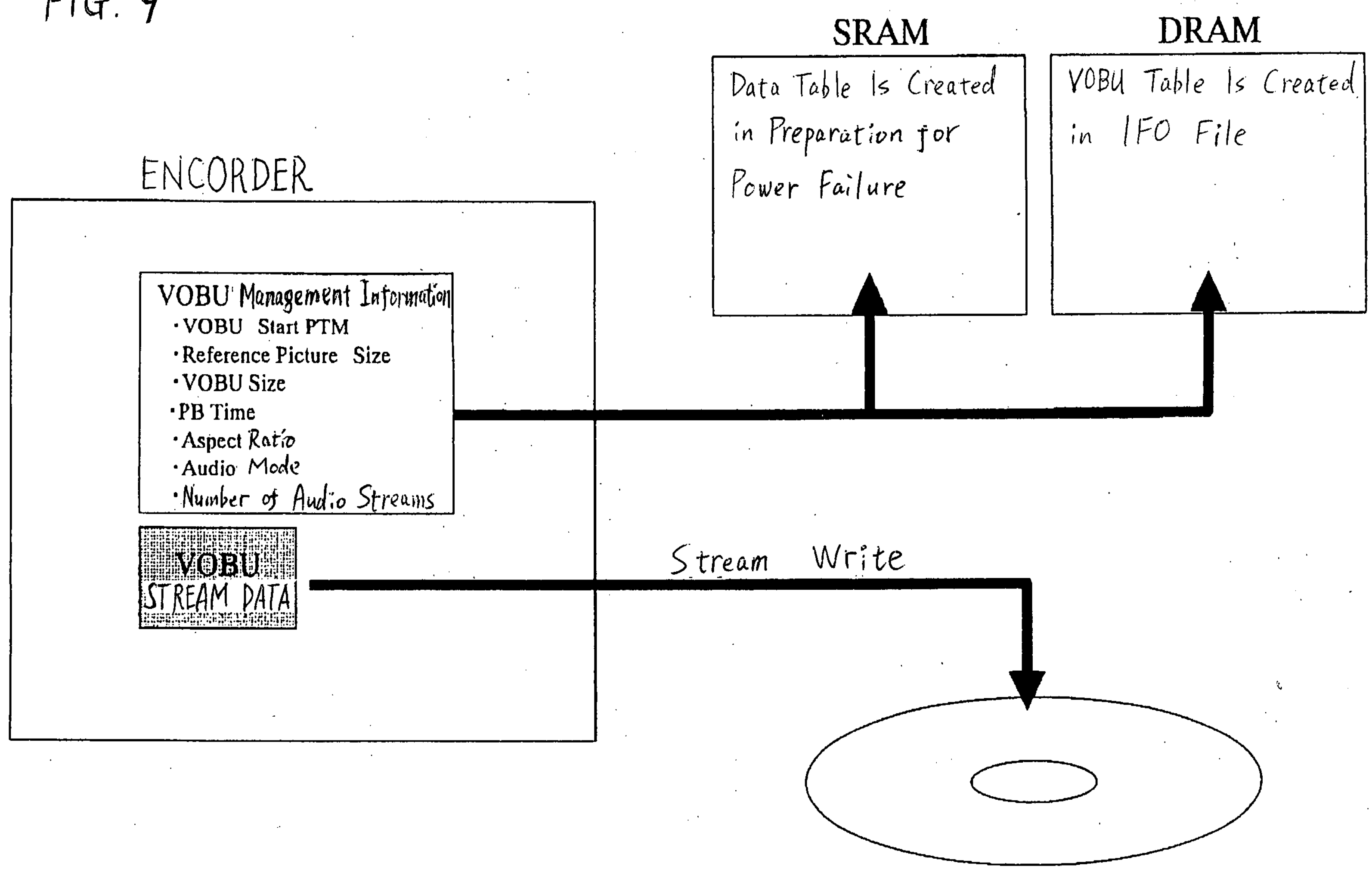
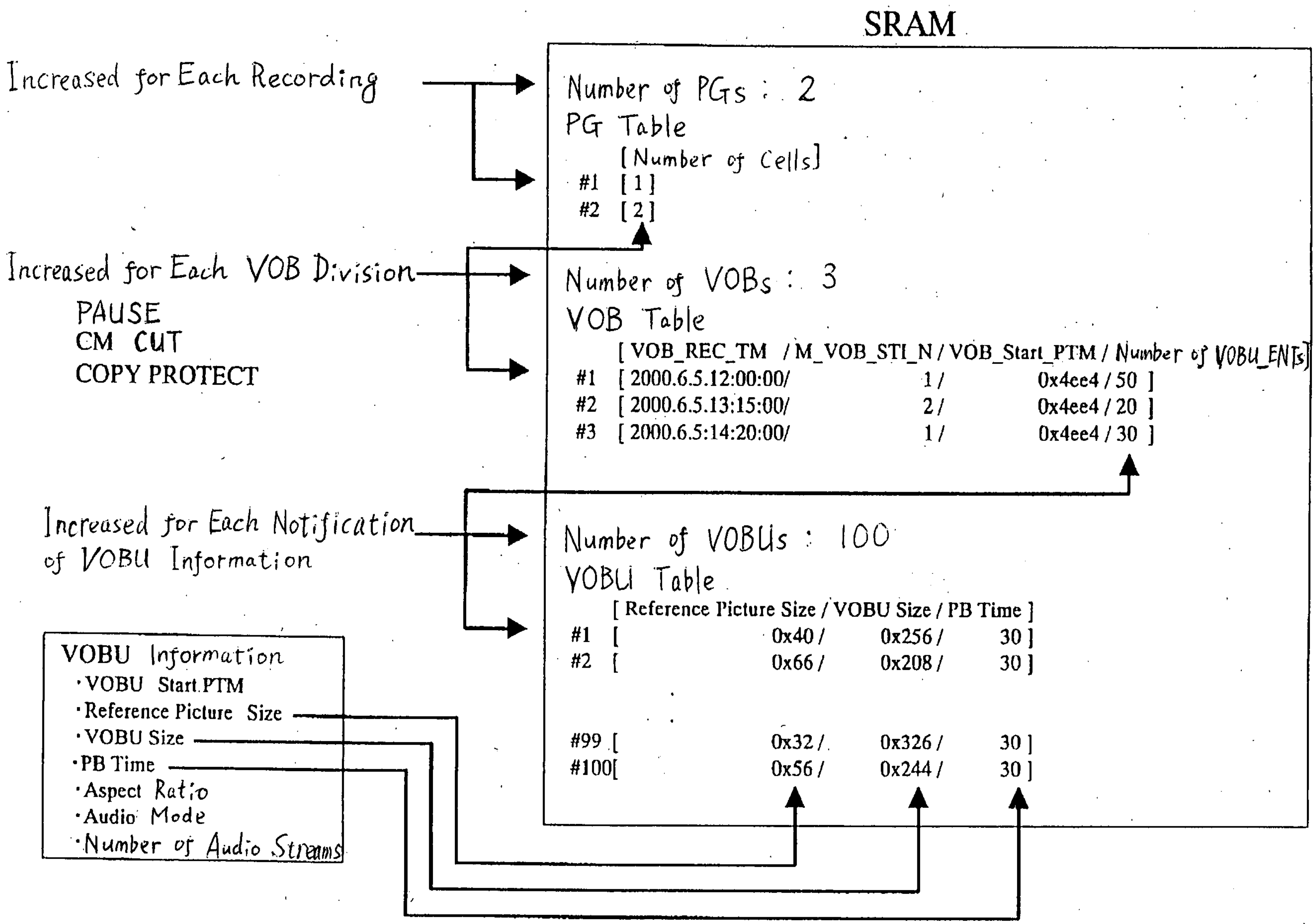

81
U/I
82
SYSTEM CONTROL UNIT
89
TIME MANAGEMENT UNIT
83
INPUT UNIT
84
ENCODER UNIT
85
OUTPUT UNIT
86
DECODER UNIT
87
TRACK BUFFER
88
DRIVE
DVD-RAM

FIG. 9



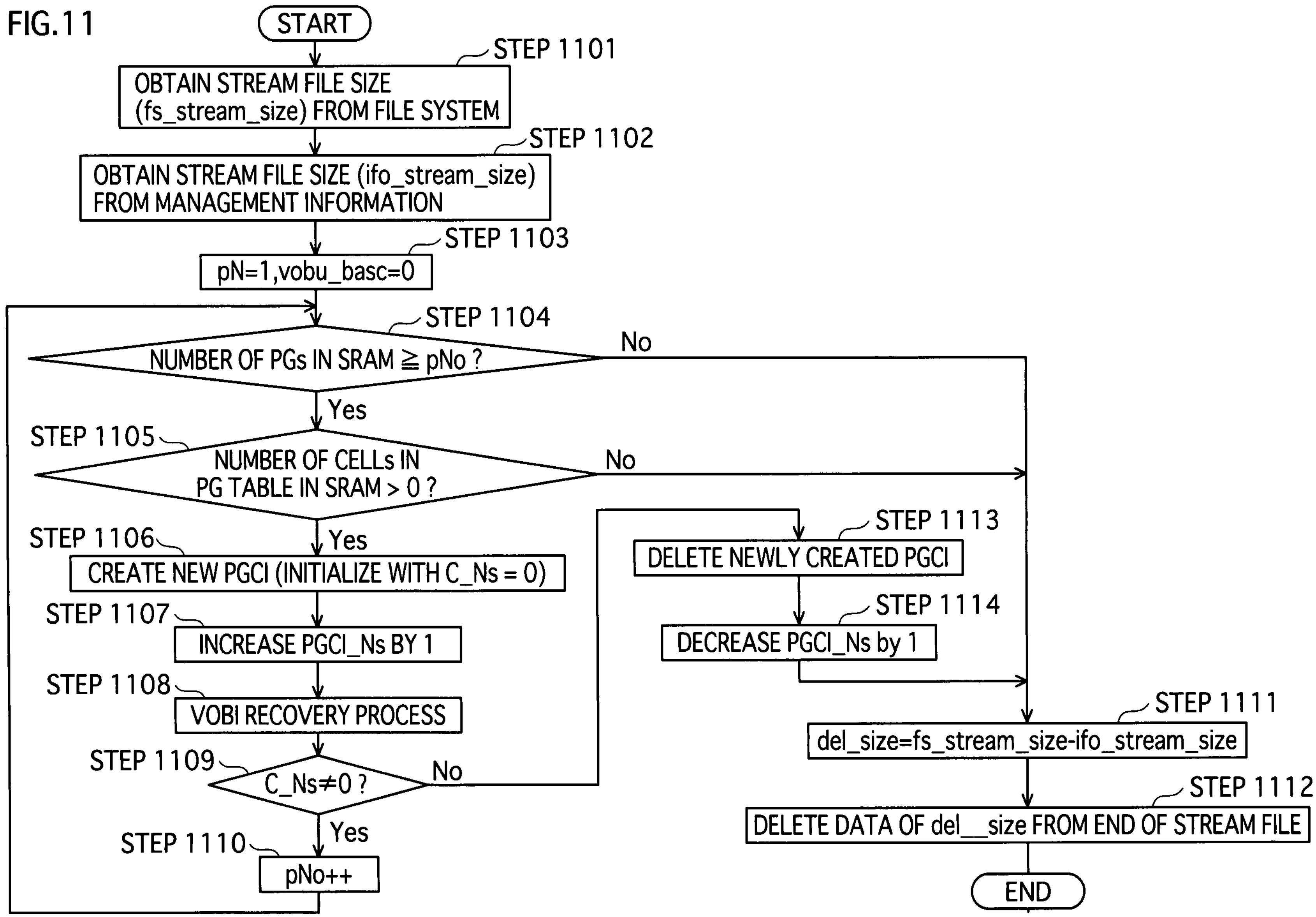
ENCORDER
VOBU Management Information
・VOBU Start PTM
・Reference Picture Size
・VOBU Size
・PB Time
・Aspect Ratio
・Audio Mode
・Number of Audio Streams
VOBU
STREAM DATA
SRAM
Data Table Is Created in Preparation for Power Failure
DRAM
VOBU Table Is Created in IFO File
Stream Write

FIG. 10



SRAM
Increased for Each Recording
Number of PGs : 2
PG Table
[Number of Cells]
#1 [1]
#2 [2]
Increased for Each VOB Division
PAUSE
CM CUT
COPY PROTECT
Number of VOBs : 3
VOB Table
[VOB_REC_TM / M_VOB_STI_N / VOB_Start_PTM / Number of VOBU_ENTs]
#1 [2000.6.5.12:00:00/ 1/ 0x4ee4 / 50]
#2 [2000.6.5.13:15:00/ 2/ 0x4ee4 / 20]
#3 [2000.6.5:14:20:00/ 1/ 0x4ee4 / 30]
Increased for Each Notification of VOBU Information
Number of VOBUs : 100
VOBU Table
[Reference Picture Size / VOBU Size / PB Time]
#1 [0x40 / 0x256 / 30]
#2 [0x66 / 0x208 / 30]
#99 [0x32 / 0x326 / 30]
#100[0x56 / 0x244 / 30]
VOBU Information
·VOBU Start PTM
·Reference Picture Size
·VOBU Size
·PB Time
·Aspect Ratio
·Audio Mode
·Number of Audio Streams

FIG.11



START
STEP 1101
OBTAIN STREAM FILE SIZE (fs_stream_size) FROM FILE SYSTEM
STEP 1102
OBTAIN STREAM FILE SIZE (ifo_stream_size) FROM MANAGEMENT INFORMATION
STEP 1103
pN=1,vobu_basc=0
STEP 1104
NUMBER OF PGs IN SRAM ≧ pNo ?
No
Yes
STEP 1105
NUMBER OF CELLs IN PG TABLE IN SRAM > 0 ?
No
Yes
STEP 1106
CREATE NEW PGCI (INITIALIZE WITH C_Ns = 0)
STEP 1107
INCREASE PGCI_Ns BY 1
STEP 1108
VOBI RECOVERY PROCESS
STEP 1109
C_Ns≠0 ?
No
Yes
STEP 1110
pNo++
STEP 1113
DELETE NEWLY CREATED PGCI
STEP 1114
DECREASE PGCI_Ns by 1
STEP 1111
del_size=fs_stream_size-ifo_stream_size
STEP 1112
DELETE DATA OF del__size FROM END OF STREAM FILE
END

FIG.12

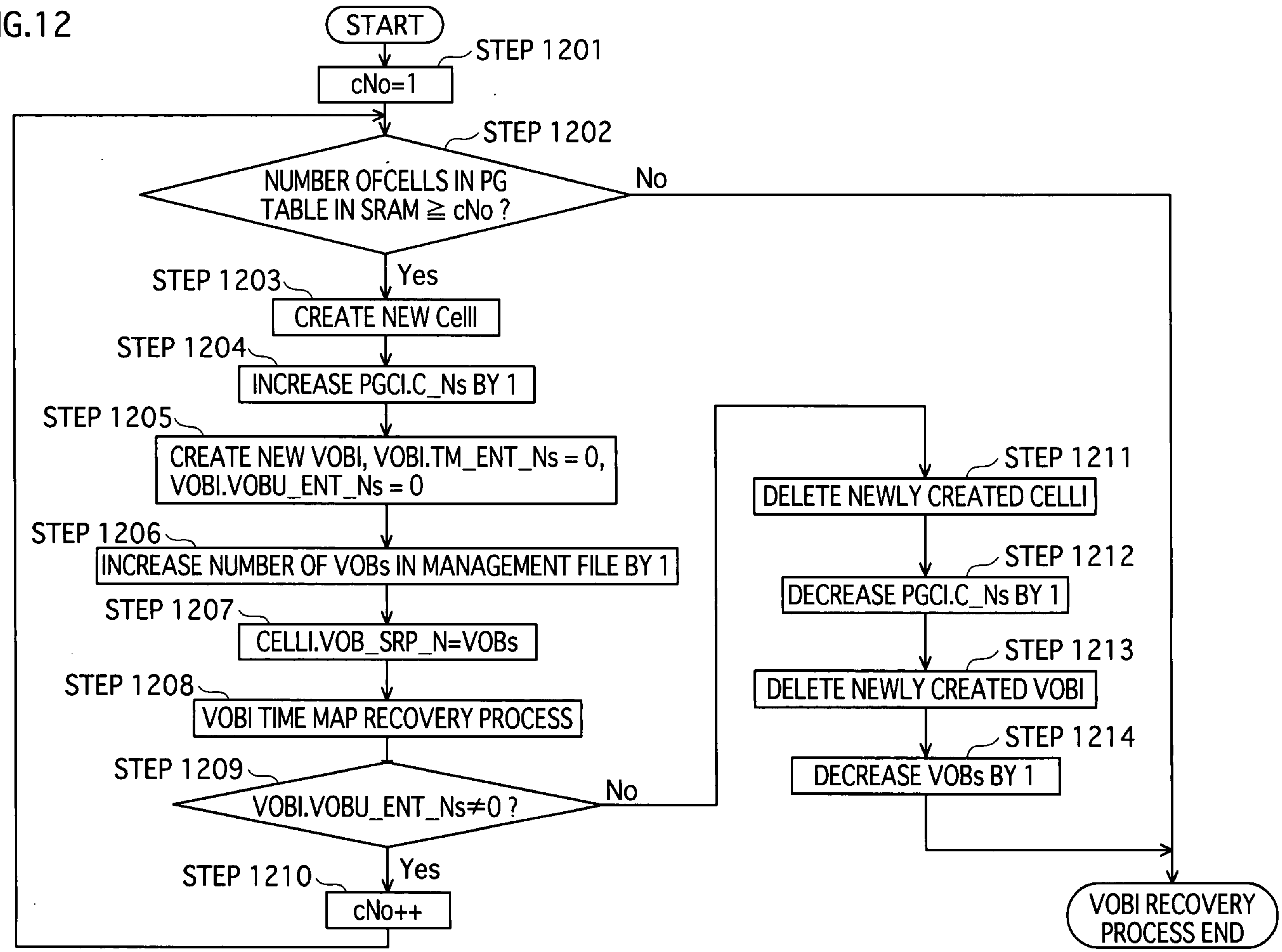

START
STEP 1201
cNo=1
STEP 1202
NUMBER OFCELLS IN PG TABLE IN SRAM ≧ cNo ?
No
Yes
STEP 1203
CREATE NEW Celll
STEP 1204
INCREASE PGCI.C_Ns BY 1
STEP 1205
CREATE NEW VOBI, VOBI.TM_ENT_Ns = 0, VOBI.VOBU_ENT_Ns = 0
STEP 1206
INCREASE NUMBER OF VOBs IN MANAGEMENT FILE BY 1
STEP 1207
CELLI.VOB_SRP_N=VOBs
STEP 1208
VOBI TIME MAP RECOVERY PROCESS
STEP 1209
VOBI.VOBU_ENT_Ns≠0 ?
No
Yes
STEP 1210
cNo++
STEP 1211
DELETE NEWLY CREATED CELLI
STEP 1212
DECREASE PGCI.C_Ns BY 1
STEP 1213
DELETE NEWLY CREATED VOBI
STEP 1214
DECREASE VOBs BY 1
VOBI RECOVERY PROCESS END

FIG.13

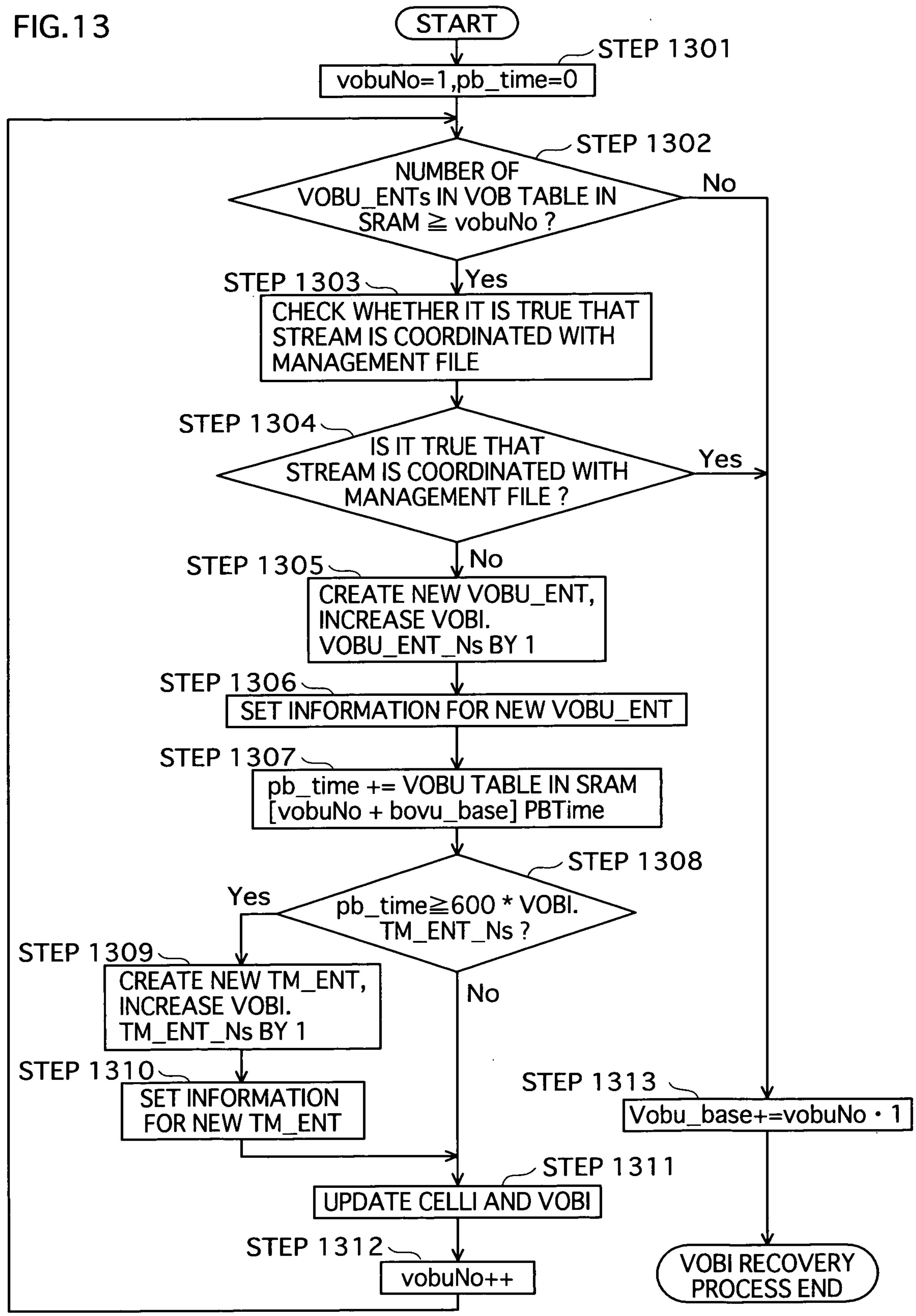
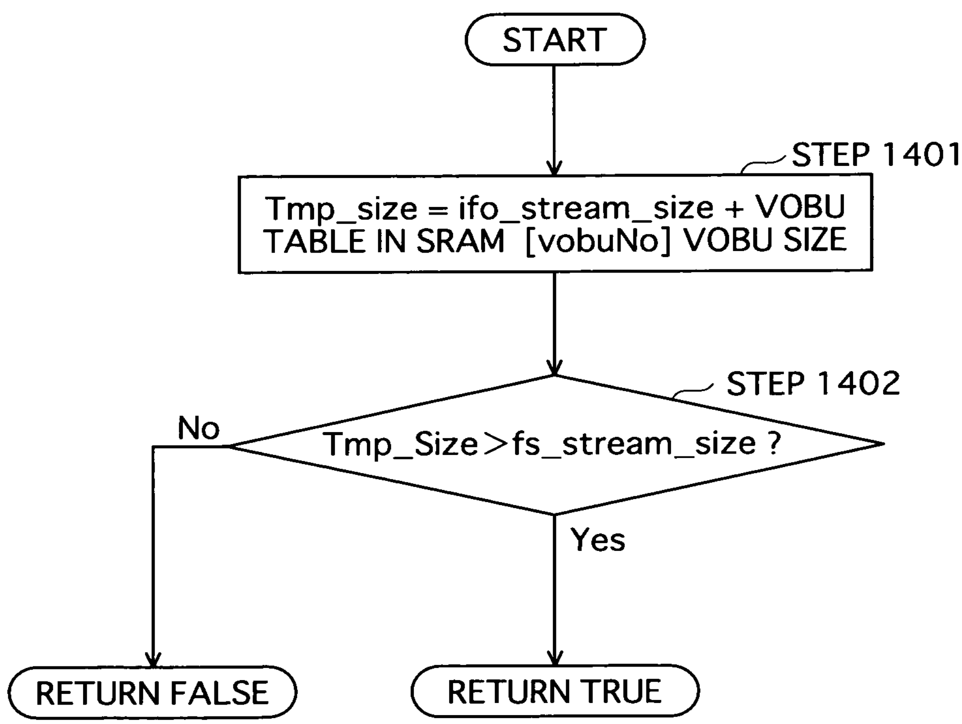

START
STEP 1301
vobuNo=1,pb_time=0
STEP 1302
NUMBER OF VOBU_ENTs IN VOB TABLE IN SRAM ≧ vobuNo ?
No
Yes
STEP 1303
CHECK WHETHER IT IS TRUE THAT STREAM IS COORDINATED WITH MANAGEMENT FILE
STEP 1304
IS IT TRUE THAT STREAM IS COORDINATED WITH MANAGEMENT FILE ?
Yes
No
STEP 1305
CREATE NEW VOBU_ENT, INCREASE VOBI. VOBU_ENT_Ns BY 1
STEP 1306
SET INFORMATION FOR NEW VOBU_ENT
STEP 1307
pb_time += VOBU TABLE IN SRAM [vobuNo + bovu_base] PBTime
STEP 1308
pb_time≧600 * VOBI. TM_ENT_Ns ?
Yes
No
STEP 1309
CREATE NEW TM_ENT, INCREASE VOBI. TM_ENT_Ns BY 1
STEP 1310
SET INFORMATION FOR NEW TM_ENT
STEP 1313
Vobu_base+=vobuNo・1
STEP 1311
UPDATE CELLI AND VOBI
STEP 1312
vobuNo++
VOBI RECOVERY PROCESS END

FIG.14


START
STEP 1401
Tmp_size = ifo_stream_size + VOBU TABLE IN SRAM [vobuNo] VOBU SIZE
STEP 1402
Tmp_Size>fs_stream_size ?
No
Yes
RETURN FALSE
RETURN TRUE